

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/512,262

REMARKS

The Office Action of June 29, 2005 has been reviewed and the comments therein were carefully considered. Claims 1-35 are pending in the instant application. Claims 1-35 stand rejected. No new matter has been introduced into the application.

Rejections under 35 USC §103

Claims 1, 2, 5, 7, 9, and 16-21 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Tada et al. (hereinafter "*Tada*"), U.S. Pat No. 5,745,745 patented 4/28/1998, in view of Fontaine et al. (hereinafter "*Fontaine*"), U.S. Pat No. 5,228,121 patented 7/13/1993. Applicants respectfully traverse these rejections.

Claim 1 recites:

A method for encoding an electronic document having markup language content, wherein the document includes at least one tag and an associated content, the method comprising the steps of:

- (a) separating the tag from the content with a separation variable;
 - (b) replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag; and
 - (c) inserting at least one flag within the tag to form an encode tag structure,
- wherein a first encoded document is formed.

Applicants submit that claim 1 is allowable over the cited art for at least the following reasons. The combination of *Tada* and *Fontaine* fails to teach at least the claimed element of "separating the tag from the content with a separation variable" as recited in step (a) of claim 1. (Emphasis Added). The Office Action states and Applicants agree that "*Tada* does not teach separating the alias from the content with a separation variable." Office Action, page 5. The Office Action relies on *Fontaine* in an attempt to disclose this missing claimed element, stating:

Fontaine teaches combining tags and information objects into a single nested structure in col. 4 line 31 – col. 5 line 44 and col. 5 line [sic]. Since, the nested structures of Fontaine logically define the order of a document and the informational objects contained within the document, the boundaries of the structure encapsulating the alias of Tada would have been separation variables separating the encode tag structure from the content.

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/512,262

Office Action, pages 5-6. However, Applicants respectfully disagree, as combining tag and information objects does not disclose the claimed feature of "separating the tag from the content with a separation variable." The boundaries of a structure may not be equated with the insertion of a separation variable. In fact, if the boundaries of Fontaine were discernable then the need for a separation variable would not be necessary. Support for the separation variable as claimed may be found on at least page 17 of Applicants specification:

At step 410, a code character is inserted to separate markup language from the actual content of the e-book file. For example, the code may be a Unicode character 0x0000. The Unicode character is inserted before and after each start and end tag. Subsequent encoding of the markup (discusses herein) should be constrained so that the Unicode character 0x0000 never occurs within the representation of a start or end-tag or elsewhere within the content, but rather occurs only as a first and last character of each start and end tag.

As neither *Tada* nor *Fontaine* disclose the claimed feature of "separating the tag from the content with a separation variable," independent claim 1 is patentable over the combination for at least this reason.

In addition, claim 1 is allowable over *Tada* and *Fontaine* for at least one additional reason. The combination of *Tada* and *Fontaine* fails to teach at least the claimed element of "replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag" as recited in step (b) of claim 1. The Office Action relies on *Tada*, contending that "*Tada* teaches replacing the tag with an alias in col. 22 lines 6-20." Office Action, page 4. Col. 22 lines 13-19 read:

A logical structure identification number corresponding to the logical structure discriminator is obtained from the correspondence table Specifically, in the example . . . the start tag '<Title>' is detected, . . . and the logical structure identification number '1' is obtained

At no point in these lines does *Tada* describe "replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag" as recited in the claim. Rather, this portion of *Tada* merely describes obtaining an identification number. It does not, however, teach or suggest the claimed element of "replacing the tag with an alias, wherein the alias is a pre-defined representation for the tag."

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/552,262

Furthermore, claim 1 is allowable over *Tada* and *Fontaine* for at least another reason. The combination of *Tada* and *Fontaine* fails to teach or suggest "inserting at least one flag within the tag to form an encode tag structure" as recited in step (c) of claim 1. The Office Action relies on *Tada*, contending that "Tada teaches in col. 22 line 24 – col. 23 line 24 inserting a control code, which is a flag, to form an encoded structure indicating whether the information contained within the tags should be searched or not." Office Action, page 4. Col. 22 lines 24-29 read:

[I]n place of the start tag, a specific control code "α" representative of the start of the logical structure is written and the obtained logical structure identification number and logical structure length are written after the control code "α". The end tag is deleted, and the control codes such as text ID and eot are written for the creation of the search database.

More explicitly, col. 22 lines 36-37 state that "the start tag is replaced by the control code α . . ." Therefore, *Tada* teaches replacing the tag with the control code. If the control code is a flag as the Office Action claims, then *Tada* does not teach or suggest "inserting at least one flag within the tag to form an encode tag structure" as recited in the claim. Rather, this portion of *Tada* would teach replacing the tag with the flag.

Finally, Applicants submit that the motivation to combine *Tada* and *Fontaine* is improper. The Office Action states:

It would have been obvious and desirable to have combined the alias and flag of Tada into a single structure in the form of a document object as taught by Fontaine so that the information would have retained a document format as is taught by Fontaine as opposed to the database format explicitly taught by Tada. Thus, by retaining a document format, the content could be manipulated and used as a document.

Applicants respectfully submit that the offered motivation cannot be found in neither *Tada* nor *Fontaine* or in the knowledge of one of ordinary skill in the art. It is respectfully submitted that it is impermissible to use the present application as a blueprint to combine references when the only suggestion can be found in the present application. Therefore, for at least the reasons describe above, Applicants respectfully submit that claim 1 is in condition for allowance. Dependent claims 2-9 which ultimately depend from claim 1 are allowable for at least the same reasons as independent claim 1.

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/512,262

Claim 16 also stands rejected under *Tada* in view of *Fontaine*. Applicants submit that this claim is allowable because the combination of *Tada* and *Fontaine*, even if proper, fails to teach or suggest the claimed features of claim 16.

Claim 16 recites:

A method for encoding an electronic document comprising the steps of:

- (a) inserting at least one code character into the electronic document to separate markup language from content;
 - (b) locating a tag within the electronic document associated with a portion of content;
 - (c) identifying a pre-defined integer alias for the tag; and
 - (d) replacing the tag with the alias,
- whereby the tag may be readily identified during run-time parsing of the document.

Applicants submit that claim 16 is allowable over the cited art for at least the following reasons. The combination of *Tada* and *Fontaine* fails to teach at least the claimed element of "inserting at least one code character into the electronic document to separate markup language from content" as recited in step (a) of claim 16. The Office Action states and Applicants agree that *Tada* does not teach separating the alias from the content with a separation variable." Office Action page 5. Thus, the Office Action relies on *Fontaine*, contending that:

Fontaine teaches combining tags and information objects into a single nested structure in col. 4 line 31 – col. 5 line 44 and col. 5 line [sic]. Since, the nested structures of *Fontaine* logically define the order of a document and the informational objects contained within the document, the boundaries of the structure encapsulating the alias of *Tada* would have been separation variables separating the encode tag structure from the content.

Office Action, pages 5-6. However, Applicants respectfully disagree, as combining tag and information objects does not disclose the claimed feature of claim 16. It is respectfully requested that the Office Action indicate what structure is being used to meet the claimed feature of "inserting at least one code character into the electronic document to separate markup language from content" as recited in the claim.

Moreover, the combination of *Tada* and *Fontaine* fails to teach at least the claimed element of "replacing the tag with an alias, whereby the tag may be readily identified during run-time parsing

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/5: 2,262

of the document” as recited in step (d) of claim 16. The Office Action relies on *Tada*, contending that “Tada teaches replacing the tag with an alias whereby the tag may be readily identified during run-time parsing of the document in col. 22 lines 6-20.” Office Action, page 5. Col. 22 lines 13-19 read:

A logical structure identification number corresponding to the logical structure discriminator is obtained from the correspondence table Specifically, in the example . . . the start tag '<Title>' is detected, . . . and the logical structure identification number '1' is obtained

As discussed with respect to claim 1, at no point in these lines does *Tada* describe “replacing the tag with an alias, whereby the tag may be readily identified during run-time parsing of the document” as recited in the claim. Rather, this portion of *Tada* merely describes obtaining an identification number. It does not, however, teach or suggest the claimed element of “replacing the tag with an alias, whereby the tag may be readily identified during run-time parsing of the document.” Therefore, for at least this additional reason, Applicants submit that claim 16 is in condition for allowance. Dependent claims 17-18 which ultimately depend from claim 16 are allowable for at least the same reason as independent claim 16.

Claim 19 also stands rejected under *Tada* in view of *Fontaine*. The Office Action alleges that “Tada teaches a tag having encoded therein a predefined integer alias for the tag in col. 1 lines 7-13, col. 21 lines 50-64, and col. 22 lines 6-20” as recited in step (a) of claim 19. Office Action, page 6. Applicants respectfully traverse this rejection.

The cited portion of *Tada* does not describe “at least one tag having encoded therein a predefined integer alias for the tag.” Rather, *Tada* describes obtaining a logical structure identification number corresponding to the logical structure discriminator. In fact, *Tada* describes that “in place of the start tag, a specific control code ‘α’ representative of the start of the logical structure is written and the obtained logical structure is written and the obtained logical structure identification number and logical structure length are written after the control code ‘α’.” It is respectfully requested that the Office Action indicate what structure is being used to meet the claimed feature of “at least one tag having encoded therein a predefined integer alias for the tag” as recited in the claim.

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/5:2,262

In addition, claim 19 is allowable for at least an additional reason. Claim 19 recited the claimed feature of "a code separating the tag from the content portion, whereby the content and markup within the document may be readily parsed at run-time." The combination of *Tada* and *Fontaine* fails to teach at least this claimed feature as discussed above. Therefore, for at least these reasons, Applicants submit that claim 19 is in condition for allowance. Claims 20-23 which ultimately depend from claim 19 are also allowable as being dependent on an allowable base claim.

Claims 10 and 11 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Tada* in view of Carus et al. (hereinafter "*Carus*"), U.S. Pat. No. 6,035,268 provisional filed 8/22/ 1996. Applicants respectfully traverse these rejections.

Claim 10 recites:

A method for pre-computing an electronic document having markup language content comprising the steps of:

- (a) identifying a tag between a left and a right term within a document;
- (b) determining whether the tag is within a single word;
- and
- (c) if the left and right terms are not part of a single word, inserting a word break flag between the left and right term, whereby a word break may be readily identified during a run-time search operation.

Applicants submit that claim 10 is allowable over the cited art for at least the following reasons. The combination of *Tada* and *Carus* fails to teach at least the claimed element of "determining whether the tag is within a single word" as recited in step (b) of claim 10. The Office Action admits and Applicants agree that "*Tada* does not teach comparing a left and right term to determine if they are part of a single word." Office Action, page 11. The Office Action relies on *Carus*, contending that "*Carus* does teach comparing a left and right term to determine if they are part of a single word . . . in col. 2 line 62 – col. 3 line 31 and col. 5 lines 51-67." Office Action, page 11. The Applicants respectfully disagree because *Carus* describes a structure in which "[t]he associated character-transition tag identifies the existence of a concatenation between successive characters, a break between successive characters, or an unknown transition between successive characters." *Carus*, col. 3 lines 28-32. Claim 10, however, claims a "method for pre-computing an electronic document having markup language content comprising the steps of . . . determining

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/5:2,262

whether the tag is within a single word," whereas *Carus* describes a structure where tags themselves identify a concatenation, break, or transition. Therefore, for at least this reason, Applicants submit that claim 10 is in condition for allowance.

In addition, claim 10 is allowable over *Tada* and *Carus* for at least one additional reason. The Office Action contends that:

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the teachings of *Tada* and *Carus* to have created the claimed invention. *Carus* notes that identifying the word breaks is a computationally expensive process in col. 2 lines 46-61. Thus, it would have been obvious and desirable to have implemented the word break identification of *Carus* in the text search improvement pre-processing so that the computationally expensive step of identifying word breaks would have been performed prior to the run-time search. Since *Tada* is also trying to pre-process text to improve run-time performance by reducing the run-time computational burden, this combination would have been very desirable to one of ordinary skill in the art at the time of the invention.

Applicants respectfully disagree. The Office Action discloses no teaching or suggestion for the combination of *Tada* and *Carus*, and it is impermissible to use the present application as a blueprint to combine references when the only suggestion can be found in the present application. Thus, Applicants submit that claim 10 is in condition for allowance for at least this additional reason. Dependent claim 11 which ultimately depends from claim 10 is allowable for at least the same reason as independent claim 10.

Claims 12 and 13 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Tada*. Applicants respectfully traverse these rejections.

Claim 12 recites:

A method for pre-computing an electronic document having markup language content comprising the steps of:

- (a) identifying a tag within a document associated with a portion of content;
- (b) determining whether the portion is to be displayed for viewing by a reading device; and
- (c) if the portion is not to be displayed for viewing, inserting a no search flag in association with the portion, whereby a no search field may be readily identified and skipped during a run-time linear search.

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/5: 2,262

Applicants submit that claim 12 is allowable over the cited art for at least the following reasons. *Tada* fails to teach at least the claimed element of "if the portion is not to be displayed for viewing, inserting a no search flag in association with the portion, whereby a no search field may be readily identified and skipped during a run-time linear search" as recited in step (c) of claim 12. The Office Action contends that "*Tada* teaches in col. 22 line 24 – col. 23 line 24 inserting a control code, which is a flag, to form an encoded structure indicating whether the information contained within the tags should be searched or not." Office Action, page 12. However, *Tada* actually compares two identification numbers to determine if the "matching process skip step is executed," and when the numbers do not match, "the text . . . is not read and is discarded." *Tada*, col. 22 line 53 – col. 23 line 3. Therefore, *Tada* teaches that it is the comparison between identification numbers that indicates whether the text is read, not the flag.

In addition, claim 12 is allowable over *Tada* for at least one additional reason. The Office Action admits and Applicants agree that "*Tada* does not teach that the no search flag is conditionally inserted based on determining whether the portion is to be displayed for viewing by the reading device." Office Action, page 12. Rather, the Office Action contends that:

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified *Tada* to have created the claimed invention. It would have been obvious and desirable to have used the search exclusion technique of *Tada* to have excluded portions which are not to be displayed by a viewing device from searching. This would have corresponded to the goal of *Tada* of improving run-time search operations as described in col. 6 lines 30 – col. 7 line 20.

Applicants respectfully disagree. The Office Action discloses no teaching or suggestion for the modification of *Tada* to insert a no search flag. Rather, *Tada* teaches and suggests the aforementioned comparison method. Thus, Applicants submit that claim 12 is in condition for allowance for at least these reasons. Dependent claim 13 which ultimately depends from claim 12 is allowable for at least the same reason as independent claim 12.

Claims 14 and 15 are rejected under 35 U.S.C. § 103(a) as being unpatentable over "Open eBook Publication Structure 1.0" (hereinafter "*Open eBook*") published 9/16/1999. Applicants respectfully traverse these rejections.

Claim 14 recites:

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/512,262

A method for pre-computing an electronic document having markup language content comprising the steps of:

- (a) identifying a Uniform Resource Locator (URL) within a document;
 - (b) searching a manifest file for a file referenced by the URL; and
 - (c) if the file is identified in the manifest file with a reference string, replacing part of the URL with the reference string and a flag for the file,
- whereby the file referenced by the URL may be readily accessed when selected during run-time.

Applicants submit that claim 14 is allowable over the cited art for at least the following reasons. The Office Action contends that "Open eBook teaches a document structure partially based on XML, HTML, and other document technologies in section 1.4 pages 3-7." Office Action, page 13. However, *Open eBook* fails to teach or suggest the claimed element of "[a] method for pre-computing an electronic document" Therefore, for at least this reason, Applicants submit that claim 14 is in condition for allowance.

In addition, claim 14 is allowable over *Open eBook* for at least one additional reason. *Open eBook* does not disclose, teach, or suggest the claimed feature of "replacing *part* of the URL with the reference string and a flag for the file." (Emphasis added). In fact this claimed feature is not disclosed in the current Office Action. It is respectfully submitted that the current Office Action merely contends that "Open eBook teaches a document structure partially based on XML, HTML, and other document technologies in section 1.4 pages 3-7. Thus, Open eBook teaches using and identifying Uniform Resource Locators (URL) within a document." Office Action, page 13. Thus, Applicants submit that claim 14 is in condition for allowance for at least this additional reason. Dependent claim 15 which ultimately depends from claim 14 is allowable for at least the same reason as independent claim 14.

Claims 24-35 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Open eBook* in view of *Tada*. Applicants respectfully traverse these rejections.

Claim 24 recites:

A computer-readable medium having stored thereon an electronic book having a file format hierarchy comprising in combination:

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/5:2,262

- (a) a root directory;
- (b) a content subdirectory linked to the root directory, the content subdirectory having nested therein at least one linked content file providing content information relating to the electronic book, wherein the content file is pre-computed and encoded to minimize computational run-time requirements.

Claim 24 recites the claimed features of "wherein the content file is pre-computed and encoded to minimize computational run-time requirements." The Office Action states and Applicants agree that *Open eBook* does not teach wherein the content file is pre-computed and encoded to minimize run-time requirements. Office Action, Page 14. The Office Action states that *Tada* teaches the advantage of decreased search time as a result of pre-computing and encoding the content file. In addition, the Office Action states:

It would have been obvious and desirable to have used the content file pre-computing and encoding as taught by *Tada* to have enabled fast run-time search operations on a *Open eBook*, when is often implemented on a low power portable reading device.

Office Action, Page 14

Applicants respectfully disagree. Neither reference discloses or suggests this motivation to combine. It respectfully submitted that it is impermissible to use the present application as a blueprint to combine references when the only suggestion can be found in the present application. Therefore, Applicants submit that independent claim 24 is allowable. Claims 25-31 depend from claim 24 and are also allowable as being dependent on an allowable base claim and further in view of additional claimed features recited therein.

Claim 32 is also stands rejected under *Open eBook* in view of *Tada*. Applicants respectfully traverse this rejection.

Claim 32 recites the claimed features of "forming a converted document. . . ." The Office Action states and Applicants agree that "*Open eBook* does not teach converting a document in a first format by processing the document to pre-compute and encode the markup language within the document. . . ." Office Action, Page 16. The Office Action states that *Tada* teaches the advantage of decreased search time as a result of pre-computing and encoding the content file. In addition, the Office Action states:

Response Dated: 09/29/2005
Response to Office Action mailed 06/29/2005

Application No. 09/512,262

It would have been obvious and desirable to have used the document and pre-computing and encoding as taught by Tada to have enabled fast run-time search operations on a Open eBook, when is often implemented on a low power portable reading device.

Office Action, Page 16

Applicants respectfully disagree. Neither reference discloses or suggests this motivation to combine. It respectfully submitted that it is impermissible to use the present application as a blueprint to combine references when the only suggestion can be found in the present application. Therefore, for at least this reason, the combination of Open eBook and *Tada* fails to support an obviousness-type rejection for independent claim 32. Claims 33-35 depend from independent claim 32 and are allowable for at least the reason discussed above with regards to independent claim 32.

Applicants respectfully submit that the instant application is in condition for allowance. Should the Examiner believe that a conversation with Applicant's representative would be useful in the prosecution of this case, the Examiner is invited and encouraged to call Applicant's representative.

Date: September 29, 2005

Respectfully submitted,

 51,393

William J. Allen
Registration No. 51,393
BANNER & WITCOFF, LTD.
10 S. Wacker Drive
Suite 3000
Chicago, IL 60606-7407
Telephone: 312-463-5000
Facsimile: 312-463-5001